INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS, VOL. 21, 409–419 (1997)

MODELLING HEAT AND MOISTURE TRANSFER IN UNSATURATED SOIL USING A FINITE DIFFERENCE SELF-IMPLICIT METHOD ON PARALLEL COMPUTERS

HYWEL RHYS THOMAS* AND CHI LEUNG WELKIN LI

Cardiff School of Engineering, University of Wales College of Cardiff, P.O. Box 917, Cardiff, CF2 1XH, Wales, U.K.

SUMMARY

A parallel numerical finite difference model, employing the self-implicit method, for coupled heat and moisture transfer in unsaturated soil is presented. The model is programmed in Occam and executed on a parallel computing network of transputers. An assessment of the model was achieved via the simulation of a laboratory experiment. A very good correlation between experimental and numerical results was obtained. Comparison of results with those obtained from a parallel explicit method is also illustrated showing no significant difference. The computational time employing the new method was, however, found to be half of that obtained using the explicit method. The computational efficiency of the approach was also found to be very high. © 1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Anal. Meth. Geomech., Vol. 21, 409–419 (1997)
(No. of Figures: 3 No. of Tables: 2 No. of Refs: 13)

Key words: parallel numerical finite difference model; heat and moisture transfer; parallel computing

INTRODUCTION

Numerical solutions of coupled heat and moisture transfer problems in unsaturated soil have been adopted for a considerable period of time.[1,2] Due to the complex and interrelated physical phenomenon involved,[3,4] the mathematical representation of this behaviour is also complicated rendering the numerical solution of the equations computationally demanding. Parallel processing is potentially one of the viable solutions to this problem. The authors[5] have previously explored this approach via the development of parallel algorithms, employing a finite difference explicit scheme. The algorithm was found to be very efficient yielding, for example, an algorithmic efficiency of 98·7 per cent for a network of nine processors.[5] However, the explicit scheme suffers from a limiting maximum time step size.

The work given here adopts an alternative scheme, a self-implicit method,[6] for parallel computational work. In this method, a solution which is still essentially explicit in nature is pursued, but additional contributions from the Taylor series expansions are included, with respect to the current finite difference node of interest.[6] The scheme remains a one step, two time

*Correspondence to: H. R. Thomas, Cardiff School of Engineering, University of Wales College at Cardiff, P.O. Box 917, Cardiff, CF2 1XH, Wales, U.K.

Contract grant sponsor: SERC; contract grant sponsor: Geotechnics steering group of the civil engineering subcommittee; contract grant number: GF/F/60403

CCC 0363–9061/97/060409–11$17.50

Received 30 June 1996
Revised 26 September 1996

level algorithm. The research is pursued here using the Occam concurrent language and a parallel computing network of transputers.

The heat and moisture transfer theory adopted is based on De Vries' approach,[3] but is developed to take into account modifications of vapour transfer characteristics in unsaturated soil.[4] Two governing differential equations, one for moisture transfer and one for energy, form a coupled theoretical formulation. The two independent variables employed in the formulation are volumetric liquid content and temperature.

The model developed is applied to a laboratory experiment of heating of medium sand.[5] The problem is set up in two-dimensional polar co-ordinates. An investigation of the performance of the self-implicit method in terms of parallel computing is presented. The accuracy of the model is also investigated by comparisons of numerical results with experimental results.

THEORETICAL FORMULATION

The governing differential moisture transfer and energy equations described by Ewen and Thomas[4] can be, respectively, written as

$$G1\,\frac{\partial \theta}{\partial t} + G2\,\frac{\partial T}{\partial t} = -\left[\frac{1}{r}\frac{\partial}{\partial r}(rU_r) + \frac{1}{r}\frac{\partial U_\phi}{\partial \phi}\right] - \left[\frac{1}{r}\frac{\partial}{\partial r}(rV_r) + \frac{1}{r}\frac{\partial V_\phi}{\partial \phi}\right] \tag{1}$$

and

$$\begin{aligned} G3\,\frac{\partial \theta}{\partial t} + G4\,\frac{\partial T}{\partial t} = {} & \frac{1}{\rho_\ell}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(rk\,\frac{\partial T}{\partial r}\right) + \frac{1}{r^2}\frac{\partial}{\partial \phi}\left(k\,\frac{\partial T}{\partial \phi}\right)\right] \\ & - \frac{C_{p\ell}}{r}\left\{\frac{\partial}{\partial r}[r(TU_r)] + \frac{\partial}{\partial \phi}(TU_\phi) - T_{\mathrm{r}}\left[\frac{\partial}{\partial r}(rU_r) + \frac{\partial U_\phi}{\partial \phi}\right]\right\} \\ & - \frac{C_{p\mathrm{v}}}{r}\left\{\frac{\partial}{\partial r}[r(TV_r)] + \frac{\partial}{\partial \phi}(TV_\phi) - T_{\mathrm{r}}\left[\frac{\partial}{\partial r}(rV_r) + \frac{\partial V_\phi}{\partial \phi}\right]\right\} \\ & - \frac{L_{\mathrm{w}}}{r}\left[\frac{\partial}{\partial r}(rV_r) + \frac{\partial V_\phi}{\partial \phi}\right] \end{aligned} \tag{2}$$

where $G1$, $G2$, $G3$ and $G4$ are the generalized storage terms, θ is the volumetric liquid content, T is the absolute temperature, T_{r} is the absolute reference temperature, r is the radius measuring from the centre, ϕ is the angle measuring counter-clockwise from the horizontal direction, t is the time, U_r and V_r are, respectively, the components of equivalent liquid velocity and equivalent vapour velocity in the r spatial co-ordinate, U_ϕ and V_ϕ are, respectively, the components of equivalent liquid velocity and equivalent vapour velocity in the ϕ spatial co-ordinate, $C_{p\ell}$ and $C_{p\mathrm{v}}$ are, respectively, the specific heat capacity of liquid water and water vapour, k is the thermal conductivity of moist soil, L_{w} is the latent heat of vaporization of liquid water and ρ_ℓ is the density of liquid water. θ can be related to U_r and U_ϕ, the components of equivalent liquid velocity, via conservation of mass considerations.[4] A full derivation of the above equations has been given previously[4] and, will not therefore be repeated here.

NUMERICAL ALGORITHM

Employing the forward difference approximation to the time derivative, upwind differencing of the second kind[7] to the convective terms in equation (2) and the central difference approximation

to other space analogues, the finite difference equations for equations (1) and (2) applicable at any node (i, j) in the finite difference grid may be written as

$$\left(G1|^n_{i,j} - \zeta\,\Delta t\,\frac{\partial \mathrm{SP}_\theta}{\partial \theta}\bigg|^n_{i,j}\right)\left(\frac{\theta|^{n+1}_{i,j} - \theta|^n_{i,j}}{\Delta t}\right) + \left(G2|^n_{i,j} - \zeta\,\Delta t\,\frac{\partial \mathrm{SP}_\theta}{\partial T}\bigg|^n_{i,j}\right)\left(\frac{T|^{n+1}_{i,j} - T|^n_{i,j}}{\Delta t}\right) = \mathrm{SP}_\theta \quad (3)$$

and

$$\left(G3|^n_{i,j} - \zeta\,\Delta t\,\frac{\partial \mathrm{SP}_E}{\partial \theta}\bigg|^n_{i,j}\right)\left(\frac{\theta|^{n+1}_{i,j} - \theta|^n_{i,j}}{\Delta t}\right) + \left(G4|^n_{i,j} - \zeta\,\Delta t\,\frac{\partial \mathrm{SP}_E}{\partial T}\bigg|^n_{i,j}\right)\left(\frac{T|^{n+1}_{i,j} - T|^n_{i,j}}{\Delta t}\right) = \mathrm{SP}_E \quad (4)$$

where the superscript n denotes the time index, the subscripts i and j indicate the spatial indices, respectively, in the r and ϕ directions, Δt is the time step size, SP_θ and SP_E are the source terms of the finite difference approximations representing the analogues of the right-hand side of equations (1) and (2), respectively, $(\partial \mathrm{SP}_\theta/\partial\theta)|^n_{i,j}$ and $(\partial \mathrm{SP}_E/\partial\theta)|^n_{i,j}$ are the rate of change of the source terms with the volumetric liquid content at the node (i, j), $(\partial \mathrm{SP}_\theta/\partial T)|^n_{i,j}$ and $(\partial \mathrm{SP}_E/\partial T)|^n_{i,j}$ are the rate of change of the source terms with the temperature at the node (i, j) and ζ is the time weighting factor taking on values between zero and unity.

The four derivative terms, $(\partial \mathrm{SP}_\theta/\partial\theta)|^n_{i,j}$, $(\partial \mathrm{SP}_E/\partial\theta)|^n_{i,j}$, $(\partial \mathrm{SP}_\theta/\partial T)|^n_{i,j}$ and $(\partial \mathrm{SP}_E/\partial T)|^n_{i,j}$, are introduced from the source terms which are approximated at time $t + \zeta\,\Delta t$ by using a Taylor series expansion in time. These four terms, which involve the rate of change of the source term with the volumetric liquid content and the temperature at the node (i, j), should enable a larger time step size to be used. This implicitness only in the dependent variables at the node (i, j) gives rise to the name, the self-implicit method. If ζ is set to zero, equations (3) and (4) represent standard explicit finite difference approximations to the moisture and energy equations. However, the calculation procedure is still explicit with factor ζ non-zero. In this paper, as a first approach, the time weighting factor is set to 0·5 i.e. the mean of the finite difference representations in the spatial derivatives on the nth and $(n + 1)$th time levels.

In equation (3) the terms $(\partial \mathrm{SP}_\theta/\partial\theta)|^n_{i,j}$ and $(\partial \mathrm{SP}_\theta/\partial T)|^n_{i,j}$ are defined fully by

$$\begin{aligned}\frac{\partial \mathrm{SP}_\theta}{\partial \theta}\bigg|^n_{i,j} = &-\frac{1}{r_{i,j}(\Delta r)^2}\left[r_{i+1/2,j}(D_{\theta\ell}|^n_{i+1/2,j} + D_{\theta\mathrm{v}}|^n_{i+1/2,j}) + r_{i-1/2,j}(D_{\theta\ell}|^n_{i-1/2,j} + D_{\theta\mathrm{v}}|^n_{i-1/2,j})\right]\\ &-\frac{1}{(r_{i,j}\Delta\phi)^2}\left(D_{\theta\ell}|^n_{i,j+1/2} + D_{\theta\mathrm{v}}|^n_{i,j+1/2} + D_{\theta\ell}|^n_{i,j-1/2} + D_{\theta\mathrm{v}}|^n_{i,j-1/2}\right)\end{aligned} \quad (5)$$

and

$$\begin{aligned}\frac{\partial \mathrm{SP}_\theta}{\partial T}\bigg|^n_{i,j} = &-\frac{1}{r_{i,j}(\Delta r)^2}\left[r_{i+1/2,j}(D_{T\ell}|^n_{i+1/2,j} + D_{T\mathrm{v}}|^n_{i+1/2,j}) + r_{i-1/2,j}(D_{T\ell}|^n_{i-1/2,j} + D_{T\mathrm{v}}|^n_{i-1/2,j})\right]\\ &-\frac{1}{(r_{i,j}\Delta\phi)^2}\left(D_{T\ell}|^n_{i,j+1/2} + D_{T\mathrm{v}}|^n_{i,j+1/2} + D_{T\ell}|^n_{i,j-1/2} + D_{T\mathrm{v}}|^n_{i,j-1/2}\right)\end{aligned} \quad (6)$$

where Δr is the spatial increment in the r direction and $\Delta\phi$ is the spatial increment in the ϕ direction, $D_{T\ell}$ is the thermal liquid diffusivity, $D_{\theta\ell}$ is the isothermal liquid diffusivity, $D_{T\mathrm{v}}$ is the thermal vapour diffusivity and $D_{\theta\mathrm{v}}$ is the isothermal vapour diffusivity.

The expression $(\partial \mathrm{SP}_E/\partial\theta)|^n_{i,j}$ in equation (4) is defined fully by

$$\frac{\partial \mathrm{SP}_E}{\partial \theta}\bigg|^n_{i,j} = \lambda_a + \lambda_b + \lambda_c + \lambda_d \quad (7)$$

with

$$\lambda_a = -\frac{C_{p\ell} T|_{i,j}^n}{r_{i,j}(\Delta r)^2}[\beta_{u1} r_{i+1/2,j} D_{\theta\ell}|_{i+1/2,j}^n + \beta_{u2} r_{i-1/2,j} D_{\theta\ell}|_{i-1/2,j}^n] - \frac{C_{p\ell} T|_{i,j}^n}{(r_{i,j}\Delta\phi)^2}(\beta_{u3} D_{\theta\ell}|_{i,j+1/2}^n + \beta_{u4} D_{\theta\ell}|_{i,j-1/2}^n) \tag{8}$$

$$\lambda_b = C_{p\ell} T_{\mathrm{r}} \left(\frac{r_{i+1/2,j} D_{\theta\ell}|_{i+1/2,j}^n + r_{i-1/2,j} D_{\theta\ell}|_{i-1/2,j}^n}{r_{i,j}(\Delta r)^2} + \frac{D_{\theta\ell}|_{i,j+1/2}^n + D_{\theta\ell}|_{i,j-1/2}^n}{(r_{i,j}\Delta\phi)^2} \right) \tag{9}$$

$$\lambda_c = -\frac{C_{p\mathrm{v}} T|_{i,j}^n}{r_{i,j}(\Delta r)^2}[\beta_{v1} r_{i+1/2,j} D_{\theta\mathrm{v}}|_{i+1/2,j}^n + \beta_{v2} r_{i-1/2,j} D_{\theta\mathrm{v}}|_{i-1/2,j}^n] - \frac{C_{p\mathrm{v}} T|_{i,j}^n}{(r_{i,j}\Delta\phi)^2}(\beta_{v3} D_{\theta\mathrm{v}}|_{i,j+1/2}^n + \beta_{v4} D_{\theta\mathrm{v}}|_{i,j-1/2}^n) \tag{10}$$

and

$$\lambda_d = (C_{p\mathrm{v}} T_{\mathrm{r}} - L_{\mathrm{w}}) \left(\frac{r_{i+1/2,j} D_{\theta\mathrm{v}}|_{i+1/2,j}^n + r_{i-1/2,j} D_{\theta\mathrm{v}}|_{i-1/2,j}^n}{r_{i,j}(\Delta r)^2} + \frac{D_{\theta\mathrm{v}}|_{i,j+1/2}^n + D_{\theta\mathrm{v}}|_{i,j-1/2}^n}{(r_{i,j}\Delta\phi)^2} \right) \tag{11}$$

β_{u1}, β_{u2}, β_{u3}, β_{u4}, β_{v1}, β_{v2}, β_{v3} and β_{v4}, in equations (8) and (10) are defined, respectively, by

$$\beta_{u1} = \frac{|U_r|_{i+1/2,j}^n| + U_r|_{i+1/2,j}^n}{2U_r|_{i+1/2,j}^n} \tag{12}$$

$$\beta_{u2} = \frac{|U_r|_{i-1/2,j}^n| - U_r|_{i-1/2,j}^n}{2|U_r|_{i-1/2,j}^n|} \tag{13}$$

$$\beta_{u3} = \frac{|U_\phi|_{i,j+1/2}^n| + U_\phi|_{i,j+1/2}^n}{2U_\phi|_{i,j+1/2}^n} \tag{14}$$

$$\beta_{u4} = \frac{|U_\phi|_{i,j-1/2}^n| - U_\phi|_{i,j-1/2}^n}{2|U_\phi|_{i,j-1/2}^n|} \tag{15}$$

$$\beta_{v1} = \frac{|V_r|_{i+1/2,j}^n| + V_r|_{i+1/2,j}^n}{2V_r|_{i+1/2,j}^n} \tag{16}$$

$$\beta_{v2} = \frac{|V_r|_{i-1/2,j}^n| - V_r|_{i-1/2,j}^n}{2|V_r|_{i-1/2,j}^n|} \tag{17}$$

$$\beta_{v3} = \frac{|V_\phi|_{i,j+1/2}^n| + V_\phi|_{i,j+1/2}^n}{2V_\phi|_{i,j+1/2}^n} \tag{18}$$

and

$$\beta_{v4} = \frac{|V_\phi|_{i,j-1/2}^n| - V_\phi|_{i,j-1/2}^n}{2|V_\phi|_{i,j-1/2}^n|} \tag{19}$$

where $|U_r|_{i+1/2,j}^n|$, $|U_r|_{i-1/2,j}^n|$, $|U_\phi|_{i,j+1/2}^n|$, $|U_\phi|_{i,j-1/2}^n|$, $|V_r|_{i+1/2,j}^n|$, $|V_r|_{i-1/2,j}^n|$, $|V_\phi|_{i,j+1/2}^n|$, and $|V_\phi|_{i,j-1/2}^n|$ are the absolute values of the velocity components.

The expression $(\partial \mathrm{SP}_E/\partial T)|_{i,j}^n$ in equation (4) is defined by

$$\left.\frac{\partial \mathrm{SP}_E}{\partial T}\right|_{i,j}^n = \varepsilon_a + \varepsilon_b + \varepsilon_c + \varepsilon_d + \varepsilon_e \tag{20}$$

with

$$\varepsilon_a = -\left(\frac{r_{i+1/2,j}k|^n_{i+1/2,j} + r_{i-1/2,j}k|^n_{i-1/2,j}}{\rho_\ell r_{i,j}(\Delta r)^2} + \frac{k|^n_{i,j+1/2} + k|^n_{i,j-1/2}}{\rho_\ell (r_{i,j}\Delta\phi)^2}\right) \tag{21}$$

$$\begin{aligned}\varepsilon_b = & -\frac{C_{p\ell}}{r_{i,j}}\left[\frac{\beta_{u1}r_{i+1/2,j}}{\Delta r}\left(D_{T\ell}|^n_{i+1/2,j}\frac{T|^n_{i,j}}{\Delta r} + U_r|^n_{i+1/2,j}\right)\right.\\ & + \frac{\beta_{u2}r_{i-1/2,j}}{\Delta r}\left(D_{T\ell}|^n_{i-1/2,j}\frac{T|^n_{i,j}}{\Delta r} - U_r|^n_{i-1/2,j}\right) + \frac{\beta_{u3}}{\Delta\phi}\left(D_{T\ell}|^n_{i,j+1/2}\frac{T|^n_{i,j}}{r_{i,j}\Delta\phi} + U_\phi|^n_{i,j+1/2}\right)\\ & \left. + \frac{\beta_{u4}}{\Delta\phi}\left(D_{T\ell}|^n_{i,j-1/2}\frac{T|^n_{i,j}}{r_{i,j}\Delta\phi} - U_\phi|^n_{i,j-1/2}\right)\right]\end{aligned} \tag{22}$$

$$\begin{aligned}\varepsilon_c = C_{p\ell}T_{\mathrm{r}}&\left[\frac{1}{r_{i,j}(\Delta r)^2}(r_{i+1/2,j}D_{T\ell}|^n_{i+1/2,j} + r_{i-1/2,j}D_{T\ell}|^n_{i-1/2,j})\right.\\ &\left. + \frac{1}{(r_{i,j}\Delta\phi)^2}(D_{T\ell}|^n_{i,j+1/2} + D_{T\ell}|^n_{i,j-1/2})\right]\end{aligned} \tag{23}$$

$$\begin{aligned}\varepsilon_d = & -\frac{C_{p\mathrm{v}}}{r_{i,j}}\left[\frac{\beta_{v1}r_{i+1/2,j}}{\Delta r}\left(D_{T\mathrm{v}}|^n_{i+1/2,j}\frac{T|^n_{i,j}}{\Delta r} + V_r|^n_{i+1/2,j}\right)\right.\\ & + \frac{\beta_{v2}r_{i-1/2,j}}{\Delta r}\left(D_{T\mathrm{v}}|^n_{i-1/2,j}\frac{T|^n_{i,j}}{\Delta r} - V_r|^n_{i-1/2,j}\right) + \frac{\beta_{v3}}{\Delta\phi}\left(D_{T\mathrm{v}}|^n_{i,j+1/2}\frac{T|^n_{i,j}}{r_{i,j}\Delta\phi} + V_\phi|^n_{i,j+1/2}\right)\\ & \left. + \frac{\beta_{v4}}{\Delta\phi}\left(D_{T\mathrm{v}}|^n_{i,j-1/2}\frac{T|^n_{i,j}}{r_{i,j}\Delta\phi} - V_\phi|^n_{i,j-1/2}\right)\right]\end{aligned} \tag{24}$$

and

$$\varepsilon_e = (C_{p\mathrm{v}}T_{\mathrm{r}} - L_{\mathrm{w}})\left(\frac{r_{i+1/2,j}D_{T\mathrm{v}}|^n_{i+1/2,j} + r_{i-1/2,j}D_{T\mathrm{v}}|^n_{i-1/2,j}}{r_{i,j}(\Delta r)^2} + \frac{D_{T\mathrm{v}}|^n_{i,j+1/2} + D_{T\mathrm{v}}|^n_{i,j-1/2}}{(r_{i,j}\Delta\phi)^2}\right) \tag{25}$$

The source terms of the finite difference approximations in equations (3) and (4) have previously been presented in detail[5] and, therefore, will not be repeated here.

PARALLEL PROCESSING

One of the major advantages of the self-implicit method, in terms of parallel computing, is that the only information required at a particular node in order to compute the value of a variable at the next time step is from that node's adjoining neighbours. In the configuration adopted here, a node's adjoining neighbours are located either within the same processor or in the adjoining processor. For the latter case, information can be concurrently transmitted through the four physical serial links connecting the processors. The calculation can then proceed concurrently in each processor. The whole computation process then moves forward to the next time step level where the parallel algorithm is again employed.

APPLICATION

Verification exercises for the numerical model were performed during each stage of development. The exercises were carried out by considering particular contributions of the model for

which solutions were known. It was found that the numerical results with the time weighting factor set to 0·5, for the range of problems considered below, agreed very well with known solutions:

(1) one-dimensional consolidation of clay and a one-dimensional horizontal infiltration problem,[8]
(2) a one-dimensional vertical flow problem, subject to concentration and flux type boundary conditions,[9]
(3) a two-dimensional transient uncoupled heat conduction problem,[10] and
(4) a one-dimensional coupled heat and mass transfer problem.[11]

The numerical model was then applied to one of a series of laboratory heating experiments carried out on medium sand.[5] Details of the experimental set-up and a complete set of properties for the medium sand have been reported elsewhere[4] and therefore only the briefest details are repeated here.

The experiment was set up by compacting a sample of sand into a cylinder. The sand was heated by an embedded circular heater passing through the centre of the cylinder. Both ends of the glass cylinder were thermally insulated and the glass cylinder was placed horizontally during the heating experiment. The sand sample was prepared initially to a 2·3 per cent volumetric liquid content and kept in an environmentally controlled room of 20°C for 14 d to allow a steady-state moisture distribution. A heating power of 53·8 W/m was then applied. Temperatures along the vertical diameter of the sample were recorded during both transient and steady-state conditions.

The region of interest was the annulus of sand between the circular heater at $r = r_1$ and the inside wall of the sample glass cylinder at $r = r_2$. Since the solution in this region is symmetrical about the vertical line passing through the centre of the circular heater, only one half of the overall domain was modelled as shown diagrammatically in Figure 1.

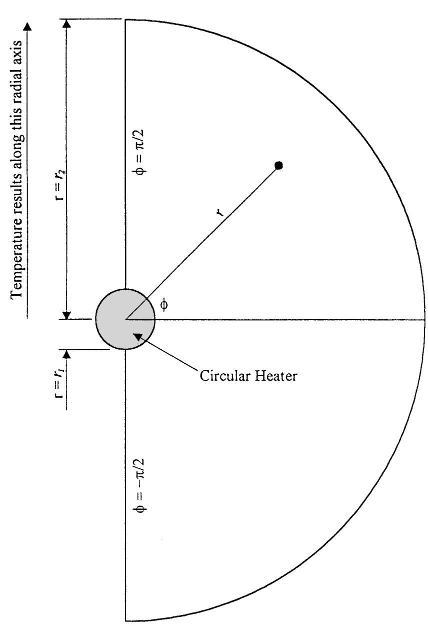

Figure 1. The modelled region

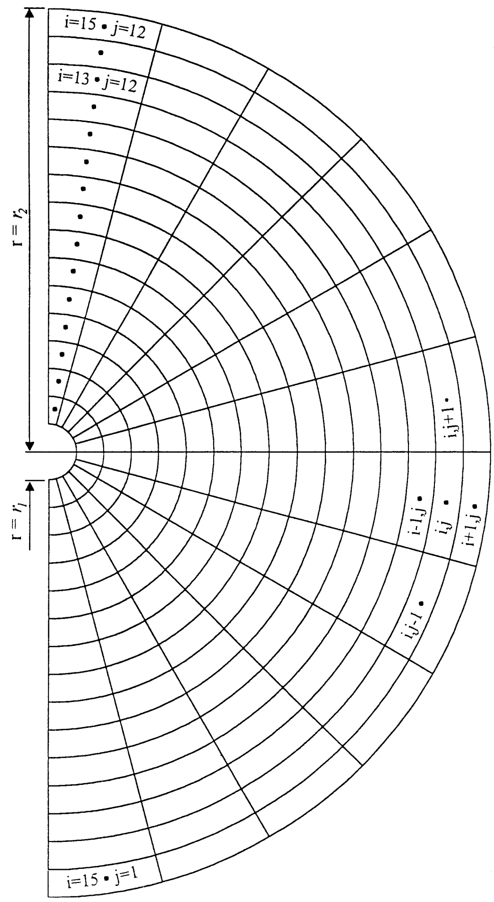

Figure 2. Finite difference grid for modelled region

The region to be modelled was divided into a 15×12 nodal array, i.e. 15 nodes in the r direction and 12 nodes in the ϕ direction as shown in Figure 2. These 15×12 nodes were equally divided and distributed across the 3×3 processor array yielding a subarray of 5×4 nodes per processor. Each subarray represented geometrically contiguous nodes. A time step size of 4 s was used.

SIMULATION RESULTS

Results of the simulated steady state temperature are plotted in Figure 3 alongside the discrete experimental data. It can be seen that an excellent match between the simulation profile and the experimentally measured data is achieved across the entire radius of the sample. Assuming a Student's t-distribution of absolute errors, it has been calculated that there is a 95 per cent probability that, at any radius, the numerical prediction is within 0·37 K of the experimental measurement. Dividing these absolute errors by the range of the experimentally measured values, this represents an error of less than 0·7 per cent. The numerical solution was also found to be conservative of mass to within 0·06 per cent.

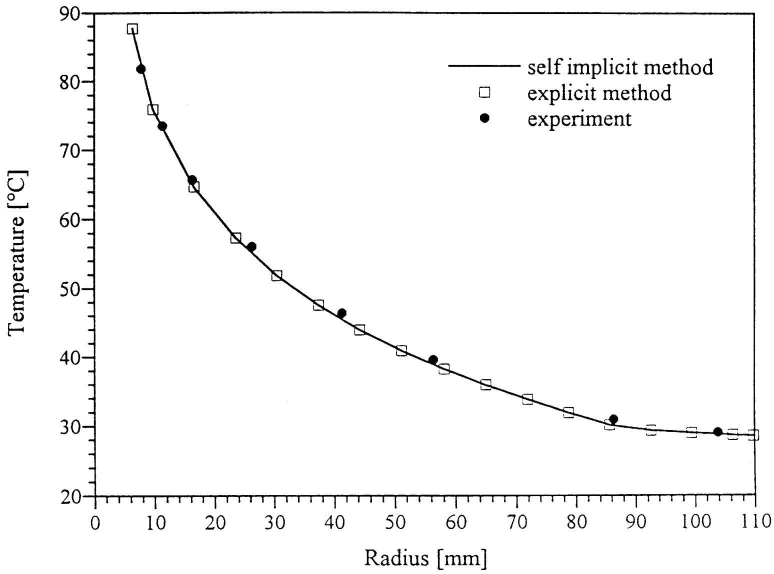

Figure 3. Steady-state temperature distributions—comparison of experimental results and numerical predictions.

The simulated steady-state temperature profile using the explicit method, i.e. by setting the time weighting factor, ζ, equal to zero, is also included in Figure 3. These results were achieved using a time step size of 2 s. It can be seen from the two numerical profiles that the two methods produce almost identical results. Again assuming a Student's t-distribution of absolute errors, it has been calculated that there is a 95 per cent probability that, at any radius, the numerical prediction for both methods is within 0·01 K. Also, again conservation of mass considerations yielded an accuracy within 0·06 per cent.

Comparing next the maximum time step size allowed by both methods for the above problem, by applying Hirt's[12] stability analysis, the maximum allowable time step size, which was found to be 2 s when employing the explicit method, increased to 4 s when using the self-implicit method. This factor of 2 increase can obviously be significant if the scheme remains efficient in parallel mode. However, the self-implicit method requires extra calculations of $\zeta\,\Delta t(\partial \mathrm{SP}_\theta/\partial\theta)|_{i,j}^n$, $\zeta\,\Delta t(\partial \mathrm{SP}_\theta/\partial T)|_{i,j}^n$, $\zeta\,\Delta t(\partial \mathrm{SP}_E/\partial\theta)|_{i,j}^n$ and $\zeta\,\Delta t(\mathrm{SP}_E/\partial T)|_{i,j}^n$ on the generalized storage terms. Doubling the time step size therefore will not necessarily halve the computational time taken. For a network of nine processor, the problem was computed to steady-state conditions of 672 h real time, which was deemed to have been reached when the neither the temperature, nor the volumetric liquid content, changed for two consecutive hours. This required a computational time of 57·6 h when employing the explicit method and 28·86 h when using the self-implicit method. It can be noted therefore that the extra calculations required to determine the generalized storage terms consumed only a very small amount of computational time. Employing the self-implicit method renders therefore a reduction of computational time by a factor of 1·996. This result is very encouraging.

Attention was then focused on the performance of the numerical algorithm in terms of parallel computational efficiency. Three tests employing the self-implicit method were performed on the above problem with the 15×12 nodal array (i.e. 180 nodal points) equally divided and loaded,

Table I. Results of algorithmic speed-up and algorithmic efficiency (9000 time steps)

Test No.	1	2	3
Total number of nodal array	15×12	15×12	15×12
Processors array	1×1	3×2	3×3
Nodal subarray per processor	15×12	5×6	5×4
Computational time (s)	13,705	2300	1543
Algorithmic speed-up	1·00	5·96	8·88
Algorithmic efficiency (%)	100·0	99·3	98·7

respectively, on a 1×1, 3×2 and 3×3 processor array. Each test was executed for a total of 9000 time steps. For the test running on one processor (1×1 processor array), no parallel overhead was incurred as the test was executed using a serial code. Table I shows the results achieved in which the algorithmic speed-up (S_a) and the algorithmic efficiency (E_a) is defined by

$$S_a = \frac{t_1}{t_p} \tag{26}$$

and

$$E_a = \frac{t_1}{Pt_p} \times 100 \text{ per cent} \tag{27}$$

where

$$t_p = t' + t_c, \tag{28}$$

P is the number of processors, t_1 is the computational time on a single processor, t_p is the computational time on P processors and t' is the central processing unit (CPU) time. t_c is the communication time which includes data transmission between processors and overhead on processor idling. The computational time, t_p, will obviously be the time it takes for the slowest processor to complete its task.

Considering the second test in Table I, the 15×12 nodal array was loaded on a 3×2 processor array yielding a subarray of 5×6 nodes per processor. From the results obtained the algorithmic speed-up and the algorithmic efficiency were calculated to be 5·96 and 99·3 per cent, respectively.

For test 3 in Table I, the 15×12 nodal array was evenly loaded on a 3×3 processor array, as 5×4 nodes per processor. The algorithmic speed-up remains very high at 8·88. The algorithmic efficiency is also maintained with a correspondingly high value of 98·7 per cent. These results illustrate that the algorithm being developed is a highly efficient parallel code.

To perform a first assessment of the performance of the numerical algorithm when applied to a larger network of processors solving a more demanding problem, the same problem was analysed on a PARAMID supercomputer consisting of i860XP vector processors, operating in tandem with T805 transputers. The software was converted to the Portland Group's ANSI FORTRAN[13] and use made of PARMACS explicit message passing libraries. The domain was discretised into a 126×126 nodal array and loaded on a 1×1, 1×21 and 1×42 processor array. In each of the three tests the problem was simulated for a total of 500 time steps. The results achieved using 1, 21 and 42 processors, respectively, are presented in Table II. It can be seen that good performance has been achieved for both tests employing 1×21 and 1×42 processor arrays. The relative speed-up for the test employing a 1×21 processor array was 18·74 yielding a 89 per cent algorithmic efficiency and the relative speed-up for the test employing a 1×42 processor array was 31·63 yielding a 75·3 per cent algorithmic efficiency. These results indicate the highly

Table II. Results of algorithmic speed-up and algorithmic efficiency (500 time steps)

Test No.	1	2	3
Total number of nodal array	126×126	126×126	126×126
Processors array	1×1	1×21	1×42
Nodal subarray per processor	126×126	126×6	126×3
Computational time (s)	2024	108	64
Algorithmic speed-up	1·00	18·74	31·63
Algorithmic efficiency (%)	100·0	89·0	75·3

efficient nature of the algorithm when distributed over a large network of processors. Further improvements in performance can be seen to be possible, when attempting to solve still larger problems, since under those conditions the computational load on the processors would be increased further, reducing the relative proportion of time spent on communication. Similarly, configuring the solution on a different form of processor array, preserving greater contiguity between processors, might also reduce communication time in relation to computing time.

CONCLUSIONS

A parallel, self-implicit finite difference numerical model has been presented for the solution of coupled heat and moisture transfer problems in unsaturated soil. The model was programmed in a concurrent language, Occam and operated on a parallel computing network of transputers.

With the time weighting factor set to 0·5, as a first approach, the model was applied to a laboratory heating experiment carried out on medium sand. An excellent match was found between the simulation results and the experimentally measured data. The temperature profile employing the self-implicit method was also compared with that obtained from the explicit method. Results from the two numerical methods showed no significant difference for this test. However, the computational time employing the self-implicit method was found to be almost half of that employing the explicit method. The algorithm was then investigated in terms of parallel computational efficiency. It was found that the algorithm developed in the model was very efficient.

ACKNOWLEDGEMENTS

The financial support of the former SERC, via research grant GF/F/60403, from the Geotechnics steering group of the civil engineering subcommittee is gratefully acknowledged.

REFERENCES

1. D. L. Slegel and L. R. Davis, 'Transient heat and mass transfer in soils in the vicinity of heated porous pipe', *ASME J. Heat Transfer*, **99,** 541–546 (1977).
2. G. L. Anders and H. S. Radhakrishna, 'Computation of temperature, field and moisture content in the vicinity of current carrying underground power cables', *IEE Proc.*, **135,** 51–62 (1988).
3. D. A. De Vries, 'Simultaneous transfer of heat and moisture in porous media', *Trans. Am. Geophys. Union*, **39,** 909–916 (1958).
4. J. Ewen and H. R. Thomas, 'Heating unsaturated medium sand', *Geotechnique* **39,** 455–470 (1989).
5. H. R. Thomas and C. L. W. Li 'Modelling transient heat and moisture transfer in unsaturated soil using a parallel computing approach', *Int. J. Numer. Anal. Methods Geomech.*, **19,** 345–366 (1995).

6. P. B. Howells and R. H. Marshall, 'An improved computer code for the simulation of solar heating systems', *Solar Energy*, **30,** 99–108 (1983).
7. P. J. Roache, *Computation Fluid Dynamics*, Hermosa Publishers, Alberquerque, NM, 1972.
8. H. R. Thomas and C. L. W. Li, 'A parallel solution of some groundwater flow problems', *Proc. 3rd Int. Symp. on Numer. Models in Geomech. (NUMOG III)*, Niagara Falls, Canada, 1989, pp. 472–480.
9. J. R. Philip, 'On solving the unsaturated flow equation: 1. The flux concentration relation', *Soil Sci*., **116,** 328–335 (1973)
10. H. S. Carslaw and J. C. Jaeger, *Conduction of Heat in Solids*, Clarendon Press, Oxford, 1959.
11. A. V. Luikov, *Heat and Mass Transfer in Capillary Porous Bodies*, Pergamon Press, Oxford, 1966.
12. C. W. Hirt, 'A heuristic stability analysis', *J. Comp. Phys.* **5,** 339–355 (1968).
13. The Portland Group, *PGF77 FORTRAN Reference Manual*, 1994.

